Brazilian Journal of Development

Enriquecimento de Base de Dorks Com Processamento de Linguagem Natural

Dorks Base Enrichment With Natural Language Processing

DOI:10.34117/bjdv6n3-085

Recebimento dos originais: 03 /02/2020
Aceitação para publicação: 06/03/2020

João Rafael Gonçalves Evangelista
Especialista em Segurança da Informação pela Universidade Nove de Julho (UNINOVE)
Instituição: Universidade Nove de Julho (UNINOVE)
Endereço: Rua Vergueiro, 235/249 - Liberdade, São Paulo - SP, 01525-000
E-mail: jrafa1607@gmail.com

Ellen Martins Lopes da Silva
Mestre em Engenharia de Produção pela Universidade Nove de Julho (UNINOVE)
Instituição: Universidade Nove de Julho (UNINOVE)
Endereço: Rua Vergueiro, 235/249 - Liberdade, São Paulo - SP, 01525-000
E-mail: ellenmartins@uni9.pro.br

Renato José Sassi
Doutor em Engenharia Elétrica pela Escola Politécnica da Universidade de São Paulo (USP)
Instituição: Universidade Nove de Julho (UNINOVE)
Endereço: Rua Vergueiro, 235/249 - Liberdade, São Paulo - SP, 01525-000
E-mail: sassi@uni9.pro.br

RESUMO

Na era digital, a informação é um dos principais ativos de uma organização, tornando-se um diferencial competitivo. Para proteger a informação, a segurança da informação dispõe de práticas para encontrar vulnerabilidades onde a informação está armazenada. Uma prática utilizada para encontrar vulnerabilidade em páginas web é o Google Hacking. O Google Hacking é uma prática de segurança da informação que utiliza dorks, strings de busca com adição ou não de operadores avançados do google. Encontra-se disponível na internet o Google Hacking Database, uma base de dados da organização Offensive Security contendo dorks testadas e validadas. Apesar da grande quantidade de dorks disponível na base, a base possui poucos atributos, fazendo-se necessário que quem a utilize, possua conhecimento prévio. Um modo de enriquecer esta base de dorks é utilizando técnicas de processamento de linguagem natural, subárea da inteligência artificial responsável por compreender, produzir e

Brazilian Journal of Development

interpretar conteúdo em linguagem humana. Diante deste cenário, o objetivo deste trabalho enriquecer base de dorks com processamento de linguagem natural no apoio em testes de segurança da informação. Como metodologia, utilizou-se pesquisa experimental com abordagem quantitativa. Os resultados mostram que o processamento de linguagem natural pode ser utilizado para enriquecer uma base de dorks.

Palavras chaves : Processamento de Linguagem Natural, Google Hacking, Google Hacking Database, Dorks, Python.

ABSTRACT

In the digital age, information is one of the main assets of an organization, becoming a competitive advantage. To protect the information, information security practices available to find vulnerabilities where the information is stored. A practice used to find vulnerability on web pages is Google Hacking. Google Hacking is an information security practice that uses back-and-forth search strings with or without the addition of advanced google operators. It is available on the Internet or on the Google Hacking Database, a database of the Offensive Security organization that contains validated dorks and tests. Despite a large number of holes available in the base, a base has few attributes, making it necessary for those who use it, have prior knowledge. One way to enrich this dorks base is to use natural language processing techniques, a subarea of artificial intelligence responsible for understanding, producing and interpreting content in human language. Given this scenario, the objective of this work is to enrich the database with natural language processing without support in information security tests. As a methodology, use experimental research with a quantitative approach. The results show that natural language processing can be used to enrich a dorks base.

Keywords: Natural Language Processing, Google Hacking, Dorks, Google Hacking Database, Python.

1 INTRODUÇÃO

O advento de novas tecnologias e suas evoluções fez com que a internet esteja em quase toda parte. As pessoas utilizam a internet para pesquisar informações, realizar compras, ver notícias, dentre outras atividades. Além disso, as pessoas conseguem compartilhar qualquer tipo de informação, independentemente dos limites de tempo e espaço (LY et al., 2018).

Com tanta informação sendo compartilhada na internet, está cada vez mais complicado mantê-las seguras. Diante do aumento da variedade de ameaças que estão surgindo, torna-se difícil proteger a informação e estimar o número e escopo dos possíveis ataques e violações contra sistemas de informação (NAARTTIJÄRVI, 2018).

A disciplina responsável por proteger a informação é a segurança da informação. Segundo a norma ISO 17799 (2005), a segurança da informação envolve a proteção da informação de diversos tipos de ameaças para garantir a continuidade do negócio, minimizar o risco ao negócio, maximizar o retorno sobre os investimentos e as oportunidades de negócio.

Para garantir a proteção da informação, a segurança da informação dispõe de padrões, métodos, processos, serviços e tecnologias para proteger não somente as informações, mas também os ativos onde a informação está armazenada (HAQAF; KOYUNCU, 2018).

Um método que pode ser utilizado para garantir a segurança das informações é descobrir as vulnerabilidades existentes onde a informação está armazenada. As vulnerabilidades representam falhas de segurança, que proporcionam riscos para a informação. (DOBROVOLJC; TRČEK; LIKAR, 2017).

Uma prática utilizada para encontrar vulnerabilidades em páginas web e sistemas online é o Google Hacking. O Google Hacking funciona como uma simples busca no google, mas utiliza uma string de busca, denominada Dork, para encontrar informações que seriam difíceis de localizar realizando buscas simples.

A organização Offensive Security possui em sua página web um banco de dados com Dorks testadas e validades, o Google Hacking Database. Apesar da grande quantidade de dorks disponíveis em sua página web, esta base possui poucos atributos, o que limita sua utilização em determinados softwares para testes de segurança e sistemas de prevenção e defesa.

Uma maneira de enriquecer a base de dados do Google Hacking Database é utilizando Processamento de Linguagem Natural. Segundo Zeroual e Lakhouaja (2018), o processamento de linguagem natural, também conhecida como linguística computacional, é um subcampo da inteligência artificial que visa aprender, compreender, reconhecer e produzir conteúdo em linguagem humana.

Diante deste cenário, este trabalho tem como objetivo enriquecer base de dorks com processamento de linguagem natural no apoio em testes de segurança da informação.

2 FUNDAMENTAÇÃO TEÓRICA

Aqui são apresentados os conceitos buscados em uma revisão da literatura sobre os temas abordados neste trabalho: *Google hacking*, *Dorks* e Processamento de Linguagem Natural.

Brazilian Journal of Development

2.1 GOOGLE HACKING

Devido ao código fonte das páginas web serem abertos, é possível encontrar vulnerabilidades em páginas web com uma maior facilidade do que em outros tipos de sistemas. Por muitas vezes, consegue-se determinar a estrutura de uma página web e a versão do código fonte apenas pesquisando por "*strings*" específicas (MANSFIELD-DEVINE, 2015).

Roy et al. (2017) apresentam uma prática utilizada em testes de segurança da informação para localizar determinados tipos de informação abertas em páginas web, apenas pesquisando por strings específicas no Google. Esta prática é conhecida como *Google Hacking* ou *Google Dorking*. A tabela 1 descreve alguns itens sobre o *Google hacking.*

Tabela 1 – Informações sobre o Google Hacking

Tabela 1 – Google Hacking. **Item**	**Descrição**
Uso do Sistema de Cache do Google	O *Google Hacking* utiliza o cache do Google para ir diretamente para um "*snapshot*" em cache de uma página web sem executar uma consulta no domínio, conseguindo assim, consultar estas páginas web sem utilizar nenhuma conexão direta com o destino.
Uso de *Dorks*	As *dorks* são *string* compostas (ou não) por operadores de busca do google. As dorks são utilizadas na prática do *Google Hacking* para encontrar vulnerabilidades em páginas web.
Descoberta de Recursos de Rede	Ao combinar operadores de busca, a prática do *Google Hacking* consegue realizar consultas DNS, obter listas de servidores e serviços disponíveis em um determinado domínio, além de poder encontrar páginas que estão conectadas à uma determinada URL.
Coleta de Arquivos	A prática do *Google Hacking* descobre não somente vulnerabilidades na estrutura de uma página web, como também pode descobrir arquivos que estão abertos ao público.
Google Hacking Database	O *Google Hacking Database* é um banco de dados com dorks que identificam vulnerabilidades. Trata-se de uma fonte autorizada utilizada em testes de segurança da informação.

Brazilian Journal of Development

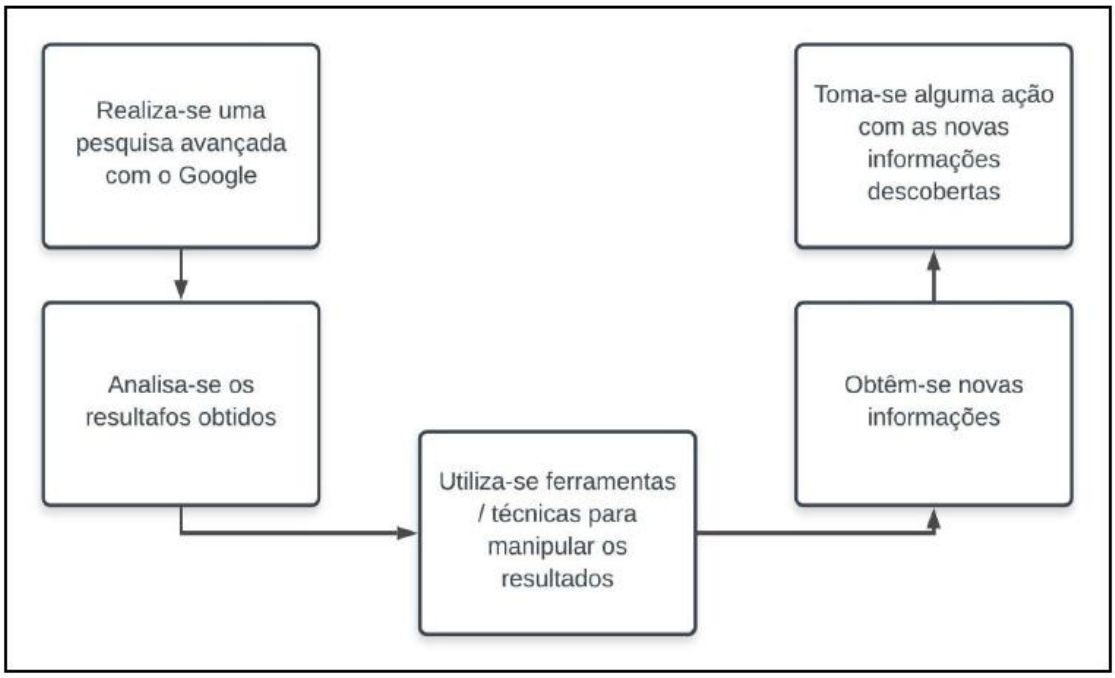
Em testes de segurança da informação, a prática do Google Hacking é utilizada na fase de reconhecimento, também conhecida como etapa preparatória. Nesta fase, procura-se reunir o máximo de informações possíveis em fontes abertas. O reconhecimento feito pelo Google Hacking é classificado como passivo, pois coleta informações sobre redes ou sistemas online sem ser invasivo (FAN; LI; ZHANG, 2018).

Em práticas maliciosas, o Google Hacking é utilizado para a obtenção de informações sensíveis. Por exemplo, se alguém mal-intencionado pesquisar com “Inurl: admin” nos mecanismos de busca, como o Google, irá encontrar painéis administrativos em muitas páginas web. (PAN et al., 2012).

Liu e Mu (2018) apontam algumas informações que podem ser coletadas praticando Google Hacking. As informações podem envolver: Nome de servidores ativos no sistema, ranges de endereços IPs, serviços online e equipamentos conectados na rede.

Pelo fato do Google Hacking ser uma prática de reconhecimento passivo, apresenta sérios desafios às contramedidas de segurança existentes, fazendo-se necessário, utilizar de outros recursos para sua detecção e prevenção (MUNIR et al., 2015). Um cenário sobre a prática do Google Hacking é apresentado na figura 1.

Figura 1 – Cenário sobre a prática do Google Hacking.



Fonte: Adaptado de Munir et al. (2015)

Brazilian Journal of Development

Pan et al. (2012) descrevem em seu estudo, categorias que podem ser utilizadas para classificar as palavras ou parâmetros que compõe as dorks. A Tabela 2 apresenta as categorias, juntamente com sua descrição e exemplos.

Tabela 2 – Exemplo de estilo de quebra de uma tabela que passa para outra página.

Categoria	**Descrição**	**Exemplos**
GRAM	Operadores Avançados do Google	Inurl, Intitle, Intext, Filetype
WEB	Vulnerabilidades em Páginas Web	Phpmyadmin, Wordpress
SCRIPT	Extensões de Páginas Web	Php, AspX, Asp, Jsp
DOC	Arquivos Desprotegidos	Doc, Pdf, Docx, Xls
DB	Arquivos de Banco de Dados	Sql, Mdb, Myd

Fonte: Adaptado de Pan et al. (2012)

2.2 PROCESSAMENTO DE LINGUAGEM NATURAL

O processamento de linguagem natural nasceu da intersecção da inteligência artificial com a sintaxe linguística nos anos 50. Suas características são distintas dos sistemas de recuperação de informação (RI), sistemas estes, que abordam a procura de diversos conteúdos, como: publicações científicas e registros de bibliotecas (HAN; KWOH, 2019).

Segundo Aggarwal, kumar e Sudarsan (2014), o processamento de linguagem natural é a área da ciência da computação preocupada em permitir que o computador consiga interpretar e entender a linguagem humana. Seu objetivo é fazer com que um computador tenha a capacidade de processar dados e instruções em linguagem natural.

O processamento de linguagem natural vem sendo utilizado na área da segurança da informação, principalmente em testes de segurança da informação, pois aumenta a eficácia da descoberta de vulnerabilidades já conhecidas em relatórios já publicados. Estes relatórios podem descrever informações como: Versões de softwares vulneráveis, tipos de vulnerabilidades em determinada aplicação e serviços vulneráveis (YOU et al., 2017).

Sun, Luo e Chen (2017) apresentam em sua revisão de literatura sobre processamento de linguagem natural, as principais bibliotecas utilizadas para implementar e desenvolver algoritmos de processamento de linguagem natural, além das principais técnicas utilizadas. A tabela 3 ilustra as principais técnicas utilizadas para processamento de linguagem natural.

Brazilian Journal of Development

Tabela 3 – Técnicas utilizadas para Processamento de Linguagem Natural

Tipo	**Técnica**
Abordagem Supervisionada	- Naive Bayes - Support Vector Machine - Max Entropy Classifiers
Abordagem Não-Supervisionada	- Algoritmos de Mineração de Associações - Métodos Baseados em Regras

Fonte: Adaptado de Sun, Luo, Chen. (2017)

A tabela 4 descreve bibliotecas que podem ser usadas para o desenvolvimento de algoritmos de processamento de linguagem natural. NLTK, OpenNLP e CoreNLP são as bibliotecas mais utilizadas, pois suportam a maioria das tarefas básicas (SUN; LUO; CHEN, 2017).

Tabela 4 – Bibliotecas para Processamento de Linguagem Natural

Biblioteca	**Linguagem**
NLTK	Python
OpenNLP	Java
CoreNLP	Java
Gensim	Python
FundanNLP	Java
LTP	C++ / Python
NiuParser	C++

Fonte: Adaptado de Sun, Luo, Chen. (2017)

3 METODOLOGIA

Aqui são apresentados o tipo de pesquisa desenvolvido, os materiais e métodos utilizados, além da condução dos experimentos computacionais realizados neste trabalho.

3.1 TIPO DE PESQUISA

Esta pesquisa é classificada como descritiva, pois envolve a aplicação de processamento de linguagem natural em uma base de *dorks*. A pesquisa descritiva é definida por Gil (2008) como pesquisa que têm como objetivo primordial a descrição das características de determinada população ou fenômeno ou o estabelecimento de relações entre variáveis.

Quanto a natureza dos dados desenvolvidos nesta pesquisa, este trabalho apresenta uma abordagem quantitativa, pois verifica se a aplicação de processamento de linguagem natural consegue enriquecer uma base de dorks.

Em relação aos procedimentos técnicos, esta pesquisa é classificada como pesquisa experimental por investigar se a aplicação de processamento de linguagem natural é capaz de enriquecer uma base de *dorks*. A pesquisa experimental consiste em determinar um objeto de estudo, selecionar as variáveis que seriam capazes de influenciá-lo, definir as formas de controle e observação dos efeitos que a variável produz no objeto (GIL, 2008).

Sobre a fundamentação teórica deste trabalho, esta pesquisa realizou uma revisão da literatura por meio das palavras-chaves: "*Natural Language Processing*", "*Google Hacking*" e "*Dorks*" nas bases de periódicos: IEEE Digital Library, ScienceDirect e EmeraldInsight.

3.2 MATERIAIS E MÉTODOS

Nesta seção são apresentados os materiais e métodos utilizados para o desenvolvimento deste trabalho. A tabela 5 descreve os materiais utilizados nesta pesquisa, juntamente com sua descrição, link de acesso e breve explicação de como foi utilizado.

Tabela 5 – Materiais utilizados no estudo.

Item	**Descrição**	**Utilização**	**Link**
Microsoft Excel 2016	Planilha Eletrônica	Utilizado para desenvolver a base de dorks	https://products.office.com/pt-br/excel
Spyder IDE	IDE para Desenvolvimento em Python	Utilizado para desenvolver o algoritmo	https://www.spyder-ide.org/
Google Hacking Database	Base de Dorks Online	Utilizado para extrair a amostra de dorks	https://www.exploit-db.com/google-hacking-database

Fonte: Autores (2020)

3.3 CONDUÇÃO DOS EXPERIMENTOS COMPUTACIONAIS

Os experimentos computacionais foram divididos em cinco fases, descritas a seguir:

a) **Fase A - Seleção das Dorks do Google Hacking Database:** Nesta primeira etapa, selecionou-se as *dorks* que foram utilizadas neste estudo.

b) **Fase B - Definição dos Atributos:** Após selecionar as *dorks*, definiu-se quais os novos atributos seriam adicionados a base para enriquecê-la.

c) **Fase C - Definição das Stopwords:** Feita a definição dos atributos, definiu-se quais seriam as stopwords que seriam removidas neste experimento.

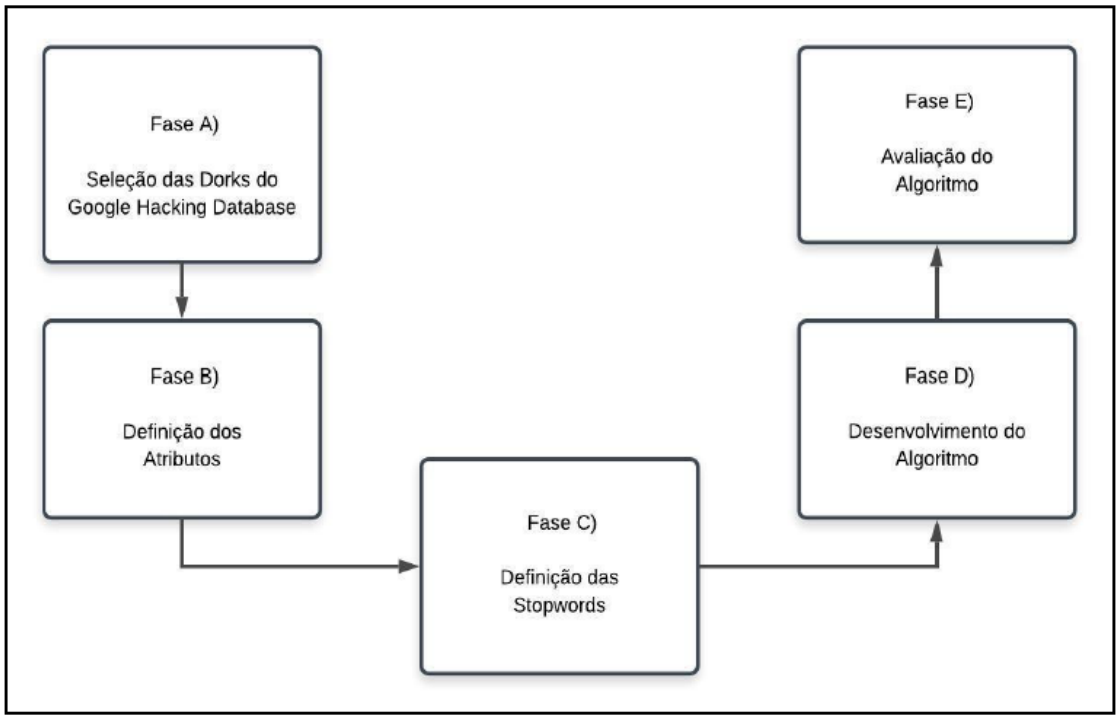
Brazilian Journal of Development

d) **Fase D - Desenvolvimento do Algoritmo:** Definido as stopwords, desenvolveu-se o algoritmo para a criação dos atributos definidos na etapa B.

e) **Fase E - Avaliação do Algoritmo:** Após desenvolver o algoritmo na etapa D, efetuou-se testes para avaliar o desempenho do algoritmo.

A figura 2 apresenta as fases dos experimentos computacionais.

Figura 2 – Condução dos Experimentos Computacionais.



Fonte: Autores (2020)

4 APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

Nesta seção, são apresentados os resultados obtidos dos experimentos computacionais.

4.1 RESULTADOS

a) Fase A - Seleção das Dorks do Google Hacking Database: Nesta etapa, selecionou-se as *dorks* do *Google Hacking Database*, base de *dorks* da *Offensive Security* disponibilizada no link: < https://www.exploit-db.com/google-hacking-database >.

Para a realização deste experimento, selecionou-se como amostra as dorks categorizadas como: **Footholds** e **Files Containing Usernames**; totalizando um total de 100

dorks selecionadas. Na primeira categoria, *Foothold*, encontram-se as *dorks* que exibem rastros deixados em páginas web, totalizando 21 dorks.

A outra categoria é a *Files Containing Usernames*, trata-se de dorks que exibem rastros de páginas ou arquivos que deixam a mostra logins de usuários de determinados tipos de sistemas. Esta categoria possui 79 Dorks.

As 100 dorks selecionadas foram inseridas em uma planilha Microsoft Excel 2016, onde utilizou-se os atributos informados pelo *Google Hacking Database*: Dork e Categoria. Além destes atributos, foi criado um atributo ID para enumerar as *dorks*, totalizando 3 atributos. O apêndice A descreve as 100 dorks extraídas nesta fase.

b) **Fase B - Definição dos Atributos:** Após selecionar as dorks, definiu-se quais os novos atributos seriam adicionados a base para enriquecê-la. O primeiro atributo definido foi o Corpus. O atributo Corpus irá possuir todas as palavras que compõe a dork, exceto os caracteres especiais.

O segundo atributo definido foi o atributo *Parameters*, onde a *dork* será dividida pelos operadores avançados de busca do google. Para isso, iniciou-se o desenvolvimento do algoritmo e criou-se uma variável denominada *Parameters*.

A variável *Parameters* trata-se de uma lista composta por sete parâmetros encontrados nas dorks extraídas na Fase A. Os Parâmetros definidos foram: ['intitle: ', 'intext: ', 'al-lintext: ', 'allintitle: ', 'ext: ', 'site: ', 'inurl: '].

c) **Fase C - Definição das Stopwords:** Feita a definição dos atributos, definiu-se quais seriam as *stopwords* que seriam removidas neste experimento.

Para isso, selecionou-se inicialmente 14 caracteres, aos quais são:

[, : ; " - = ‘ ! ? ‘’ ‘ ’ ( ) @].

Mas após a execução do algoritmo, foi necessário incluir mais 6 *stopwords*, totalizando 20 *stopwords*. As 20 *stopwords* foram:

[, : ; " - = ‘ ! ? ‘’ ‘ ’ ( ) @ ~ / | ' * ``].

d) **Fase D - Desenvolvimento do Algoritmo:** Definido as stopwords, desenvolveu-se o algoritmo para a criação dos atributos definidos na etapa B.

Para iniciar o algoritmo, importou-se as bibliotecas *pandas* e *numpy* para cálculos matemáticos e manipulação de matrizes, juntamente com a biblioteca nltk para processamento de linguagem natural.

Em seguida importou-se a planilha criada na etapa A com as *dorks* para uma variável denominada *dataframe*. Feito isso, criou-se os dois atributos na variável dataframe, o atributo

Brazilian Journal of Development

corpus e o atributo *parameters*. Após criar estes atributos, preencheu-se os valores nulos do *dataframe* com o valor "*missing*" retirando assim, todos os valores nulos do *dataframe*.

Finalizado a estruturação da variável *dataframe*, começou-se a testar comandos para conseguir preencher corretamente os dois novos atributos inseridos no dataframe. A tabela 6 apresenta o resultado dos testes.

Tabela 6 – Testes com as Dorks

Comando	**Objetivo**	**Corpus**	**Parameters**
split("")	Separar a dork a cada aspas (") encontrada.	X	X
split(':')	Separar a dork a cada dois pontos (:) encontrado.	X	X
split(' ')	Separar a dork a cada espaço (' ') encontrado	X	✅
Tokenização com NLP	Separação da dork em tokens (fragmentação)	X	X
Tokenização com NLP + Remoção das Stopwords	Separação da dork em tokens (fragmentação) + Remoção das stopwords	✅	X

Assim, para o acréscimo do atributo *parameters*, utilizou-se o comando Split com valores em branco nas dorks (' '). Já para o atributo *corpus*, utilizou-se a Tokenização por processamento de linguagem natural em adição a remoção das stopwords.

e) **Fase E - Avaliação do Algoritmo:** Após desenvolver o algoritmo na etapa D, efetuou-se testes para avaliar o desempenho do algoritmo.

Para isso, dividiu-se a execução do algoritmo em 3 fases. A primeira fase contendo somente as dorks da categoria *Files Containing Username*. A segunda fase foi executada com 50% da base, com 21 dorks da categoria *File Containing Username* e 29 dorks da Categoria *Foothold*. E por fim, testou-se o algoritmo em 100% da base, totalizando 100 dorks.

Em todas as 3 fases, o algoritmo conseguiu atribuir valores para os novos atributos em 100% das dorks oferecidas quando se utilizou a Tokenização por processamento de linguagem natural junto a remoção das stopwords e split com o valor (' ').

5 CONCLUSÕES

Este trabalho abordou o enriquecimento de base de *dorks* com processamento de linguagem natural. O desenvolvimento do algoritmo junto a aplicação do processamento de

linguagem natural por meio da Tokenização e remoção de stopwords conseguiu enriquecer a base de dorks, gerando dois novos atributos: *corpus* e *parameters*.

Com estes novos atributos, torna-se possível utilizar esta base de *dorks* em testes de segurança automatizados, por exemplo em frameworks de Inteligência de Fontes Abertas. Além disso, a base de dorks enriquecida também pode ser utilizada para se criar regras em sistemas como Firewalls, IPS e IDS para que possam detectar buscas maliciosas dentro de um domínio.

Para trabalhos futuros, recomenda-se a seleção de toda a base do *Google Hacking Database* e a aplicação de técnicas de processamento de linguagem natural para classificação e agrupamento das *dorks*, gerando assim, novos atributos para a base.

AGRADECIMENTOS

Agradeço à Universidade Nove de Julho - UNINOVE pelo apoio à pesquisa.

REFERÊNCIAS

AGGARWAL, Shivam; KUMAR, Vishal; SUDARSAN, S. D. **Identification and detection of phishing emails using natural language processing techniques**. In: Proceedings of the 7th International Conference on Security of Information and Networks. ACM, p. 217. 2014. https://doi.org/10.1145/2659651.2659691.

DOBROVOLJC, Andrej; TRČEK, Denis; LIKAR, Borut. **Predicting Exploitations of Information Systems Vulnerabilities Through Attackers' Characteristics**. IEEE Access, p. 26063-26075, 2017. https://doi.org/10.1109/ACCESS.2017.2769063.

FAN, Youping; LI, Jingjiao; ZHANG, Dai. **A Method for Identifying Critical Elements of a Cyber-Physical System Under Data Attack**. IEEE Access, v. 6, p. 16972-16984, 2018. https://doi.org/10.1109/ACCESS.2018.2812812.

GIL, Antônio Carlos. **Métodos e técnicas de pesquisa social**. 6. ed. Ediitora Atlas SA, 2008.

HAN, Xu; KWOH, Chee K. **Natural Language Processing Approaches in Bioinformatics**. Encyclopedia of Bioinformatics and Computational Biology. v. 1. p. 561-574. 2019. https://doi.org/10.1016/B978-0-12-809633-8.20463-9.

HAQAF, Husam; KOYUNCU, Murat. **Understanding key skills for information security managers**. Interna-tional Journal of Information Management, v. 43, p. 165-172, 2018. https://doi.org/10.1016/j.ijinfo-mgt.2018.07.013.

ISO, ABNT NBR. IEC 17799: 2005: **Tecnologia da informação–Técnicas de segurança–Código de prática para a gestão da segurança da informação**. Rio de Janeiro: ABNT, 2006.

LIU, Yixian; MU, Dejun. **A Network Security Situation Awareness Model Based on Risk Assessment**. In: The Euro-China Conference on Intelligent Data Analysis and Applications. Springer. p. 17-24. 2018. https://doi.org/10.1007/978-3-030-03766-6_3.

LY, Pham Thi Minh; LAI, Wen-Hsiang; HSU, Chiung-Wen; SHIH, Fang-Yin. **Fuzzy AHP analysis of Internet of Things (IoT) in enterprises**. Technological Forecasting and Social Change, v. 136, p. 1-13, 2018. https://doi.org/10.1016/j.techfore.2018.08.016.

MANSFIELD-DEVINE, Steve. **Taking responsibility for security**. Computer Fraud & Security, v. 2015, n. 12, p. 15-18, 2015. https://doi.org/10.1016/S1361-3723(15)30112-3.

MUNIR, Rashid; MUFTI, Muhammad Rafiq; AWAN, Irfan; HU, Yim Fun; DISSO, Jules Pagna. **Detection, mitigation and quantitative security risk assessment of invisible attacks at enterprise network**. In: 2015 3rd International Conference on Future Internet of Things and Cloud. IEEE, p. 256-263. 2015. https://doi.org/10.1109/FiCloud.2015.24.

NAARTTIJÄRVI, Markus. **Balancing data protection and privacy–The case of information security sensor systems**. Computer Law & Security Review, v. 34, p. 1019-1038. 2018. https://doi.org/10.1016/j.clsr.2018.04.006.

PAN, Daoxin; BAI, Wei; ZHANG, Siyu; ZOU, Futai. **Detecting Malicious Queries from Search Engine Trae-fic.** In: 2012 8th International Conference on Wireless Communications, Networking and Mobile Computing. IEEE, p. 1-4. 2012. https://doi.org/10.1109/WiCOM.2012.6478492.

ROY, Ahana; MEIJA, Louis; HELLING, Paul; OLMSTED, Aspen. **Automation of cyber-reconnaissance: A Java-based open source tool for information gathering**. In: ICITST - International Conference for Internet Technology and Secured Transactions. p. 424-426. 2017. https://doi.org/10.23919/ICITST.2017.8356437.

SUN, Shiliang; LUO, Chen; CHEN, Junyu. **A review of natural language processing techniques for opinion mining systems**. Information Fusion, v. 36, p. 10-25, 2017. https://doi.org/10.1016/j.inffus.2016.10.004.

YOU, Wei; ZONG, Peiyuan; CHEN; Kai, WANG, XiaoFeng; LIAO, Xiaojing; BIAN, Pan; LIANG, Bin. Sem-Fuzz: Semantics-based Automatic Generation of Proof-of-Concept

Exploits. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, p. 2139-2154. 2017.

ZEROUAL, Imad; LAKHOUAJA, Abdelhak. **Data science in light of natural language processing: An over-view.** Procedia Computer Science, v. 127, p. 82-91, 2018. https://doi.org/10.1016/j.procs.2018.01.101.

Brazilian Journal of Development

Apêndice A

Neste apêndice, é descrito as dorks utilizadas neste experimento. Cada dork possui um ID e uma Categoria.

ID	Dork	Category
1	"username.xlsx" ext:xlsx	Files Containing Usernames
2	inurl:/_layouts/mobile/view.aspx?List=	Files Containing Usernames
3	authentication failure; logname=" ext:log	Files Containing Usernames
4	inurl:/profile.php?lookup=1	Files Containing Usernames
5	intext:"root:x:0:0:root:/root:/bin/bash" inurl:*=/etc/passwd	Files Containing Usernames
6	inurl:"/root/etc/passwd" intext:"home/*:"	Files Containing Usernames
7	site:extremetracking.com inurl:"login="	Files Containing Usernames
8	intext:"SteamUserPassphrase=" intext:"SteamAppUser=" -"username" -"user"	Files Containing Usernames
9	inurl:root.asp?acs=anon	Files Containing Usernames
10	filetype:conf inurl:proftpd.conf -sample	Files Containing Usernames
11	filetype:log username putty	Files Containing Usernames
12	filetype:reg reg +intext:"internet account manager"	Files Containing Usernames
13	filetype:reg reg HKEY_CURRENT_USER username	Files Containing Usernames
14	+intext:"webalizer" +intext:"Total Usernames" +in-text:"Usage Statistics for"	Files Containing Usernames
15	inurl:php inurl:hlstats intext:"Server Username"	Files Containing Usernames
16	"index of" / lck	Files Containing Usernames
17	index.of perform.ini	Files Containing Usernames
18	inurl:admin filetype:asp inurl:userlist	Files Containing Usernames
19	inurl:admin inurl:userlist	Files Containing Usernames
20	intitle:index.of .bash_history	Files Containing Usernames
21	intitle:index.of .sh_history	Files Containing Usernames
22	intext:"M3R1C4 SHELL BACKDOOR"	Foothold
23	"index of" /wp-content/uploads/shell.php	Foothold
24	"File Manager - Current disk free"	Foothold
25	inurl: "Mister Spy" | intext:"Mister Spy & Souheyl Bypass Shell"	Foothold
26	inurl:"/tiny_mce/plugins/ajaxfi-lemanager/inc/data.php" | inurl:"/tiny_mce/plugins/ajaxfilemanager/ajax_cre-ate_folder.php" -github	Foothold
27	intitle:Upload inurl:/cgi-bin/filechucker.cgi	Foothold
28	intitle:"Installing TYPO3 CMS"	Foothold
29	inurl:/install/stringnames.txt	Foothold
30	intitle:"Solr Admin" "Solr Query Syntax"	Foothold
31	ext:jsp intext:"jspspy" intitle:"Jspspy web~shell V1.0"	Foothold
32	"Sorting Logs:" "Please enter your password" "Powered By" -urlscan -alamy	Foothold

Brazilian Journal of Development

33	intitle:"Authorization" "TF" inurl:"admin.php"	Foothold
34	ext:php intext:"-rwxr-xr-x" site:.in	Foothold
35	intitle:index of intext:@WanaDecryptor@.exe	Foothold
36	intitle:index of intext:wncry	Foothold
37	inurl:"go.cgi?url="	Foothold
38	"WHMCS Auto Xploiter"	Foothold
39	"El Moujahidin Bypass Shell" ext:php	Foothold
40	intitle:"Priv8 Mailer Inbox 2015" ext:php	Foothold
41	(ext:php) (inurl:/wp-content/uploads/AAPL/loaders/)	Foothold
42	inurl:?filesrc=**** ~"Current" ~"asp"	Foothold
43	inurl:/\\filesrc=**** ~"Current" ~":/" ~"upload"	Foothold
44	"PHP Mailer" "priv8 Mailer" ext:php	Foothold
45	Meg4-Mail ext:php	Foothold
46	"PHP eMailer is created by" ext:php	Foothold
47	"File Manager Version 1.0" "Coded By"	Foothold
48	inurl:"html/js/editor/ckeditor/"	Foothold
49	"You have selected the following files for upload (0 Files)."	Foothold
50	intitle:"nstview v2.1:: nst.void.ru" | in-text:"nsTView v2.1 :: nst.void.ru. Password: Host:"	Foothold
51	filetype:php intext:Your Email: intext:Your Name: in-text:Reply-To: intext:mailer	Foothold
52	intitle:"Hamdida X_Shell Backd00r"	Foothold
53	"Fenix Final Version v2.0" filetype:php	Foothold
54	intitle:Automatic cPanel Finder/Cracker | 3xp1r3 Cyber Army	Foothold
55	(intitle:"phpshell" OR intitle:"c99shell" OR inti-tle:"r57shell" OR intitle:"PHP Shell " OR inti-tle:"phpRemoteView") `rwx` "uname"	Foothold
56	intitle: "phpshell" "Php Safe-Mode Bypass"	Foothold
57	intitle:"Shell I" inurl:revslider inurl:error.php inurl:cmd	Foothold
58	inurl:revslider inurl:temp inurl:update_extract inurl:sym1	Foothold
59	intext:"Sw Bilgi" ext:php	Foothold
60	intext:Developed By Black.Hack3r ext:php	Foothold
61	ext:php intitle:"b374k"	Foothold
62	ext:aspx intitle:aspxspy	Foothold
63	intitle:"WSO " ext:php intext:"server ip" 2015 in-text:" [home]"	Foothold
64	crime24 stealer ext:txt	Foothold
65	intext:"Please select file to upload:" ext:php	Foothold
66	intext:"Thehacker - Agd_Scorp - BLaSTER - Cr@zy_King - KinSize - JeXToXiC - s3f4 - rx5"	Foothold
67	inurl:sh3llZ/c99/	Foothold
68	intitle:SN0X SHELL: WEEEEEEEEEEEEEEEED	Foothold

Brazilian Journal of Development

69	ext:asp intext:Smart.Shell 1.0 BY P0Uy@_$3r\/3R -	Foothold
70	intitle:"=[1n73ct10n privat shell]="	Foothold
71	intitle:"WSO 2.4" [Sec. Info], [Files], [Console], [Sql], [Php], [Safe mode], [String tools], [Bruteforce], [Network], [Self remove]	Foothold
72	filetype:php intext:"!C99Shell v. 1.0 beta"	Foothold
73	intitle:"uploader by ghost-dz" ext:php	Foothold
74	inurl:1337w0rm.php intitle:1337w0rm	Foothold
75	Re: intitle:Priv8 SCR	Foothold
76	intitle:C0ded By web.sniper	Foothold
77	Re: inurl:"r00t.php"	Foothold
78	inurl:"amfphp/browser/servicebrowser.swf"	Foothold
79	allintext:"fs-admin.php"	Foothold
80	(intitle:"SHOUTcast Administrator")|(intext:"U SHOUT-cast D.N.A.S. Status")	Foothold
81	(intitle:"WordPress Ã¢â‚¬Â° Setup Configuration File")|(inurl:"setup-config.php?step=")	Foothold
82	"index of /" (upload.cfm | upload.asp | upload.php | upload.cgi | upload.jsp | upload.pl)	Foothold
83	"Please re-enter your password It must match exactly"	Foothold
84	inurl:"tmtrack.dll?"	Foothold
85	inurl:polly/CP	Foothold
86	intitle:"net2ftp" "powered by net2ftp" inurl:ftp OR intext:login OR inurl:login	Foothold
87	intitle:MyShell 1.1.0 build 20010923	Foothold
88	intitle:"YALA: Yet Another LDAP Administrator"	Foothold
89	intitle:"ERROR: The requested URL could not be re-trieved" "While trying to retrieve the URL" "The fol-lowing error was encountered:"	Foothold
90	inurl:"phpOracleAdmin/php" -download -cvs	Foothold
91	PHPKonsole PHPShell filetype:php -echo	Foothold
92	filetype:php HAXPLORER "Server Files Browser"	Foothold
93	inurl:ConnectComputer/precheck.htm | inurl:Remote/lo-gon.aspx	Foothold
94	(inurl:81/cgi-bin/.cobalt/) | (intext:"Welcome to the Cobalt RaQ")	Foothold
95	intitle:"Web Data Administrator - Login"	Foothold
96	"adding new user" inurl:addnewuser -"there are no do-mains"	Foothold
97	"Powered by PHPFM" filetype:php -username	Foothold
98	intitle:"PHP Shell *" "Enable stderr" filetype:php	Foothold
99	"=+htpasswd +WS_FTP.LOG filetype:(log)"	Foothold
100	intitle:admin intitle:login	Foothold

Brazilian Journal of Development